SQL Database

Upon completion of this module, a student will be able to

- explain what an SQL Database is
- retrieve listed data from the database
- retrieve filtered data from the database
- add data to the database
- remove data from the database
- define an SQL Schema

A Student Can

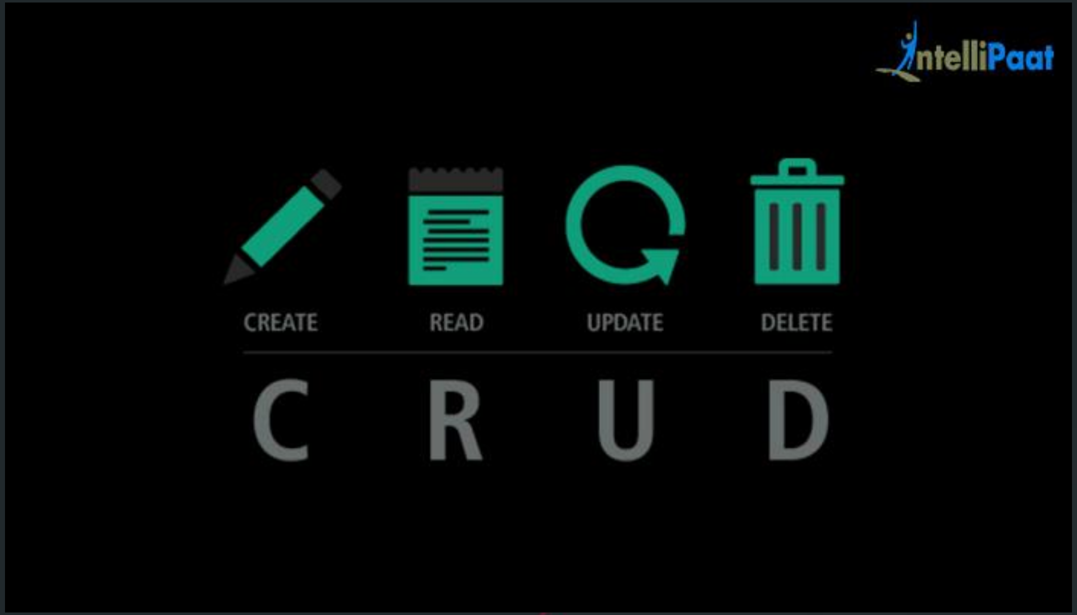
explain what an SQL Database is

SQL Database

- Structured Query Language
- Relational Database

CRUD-L

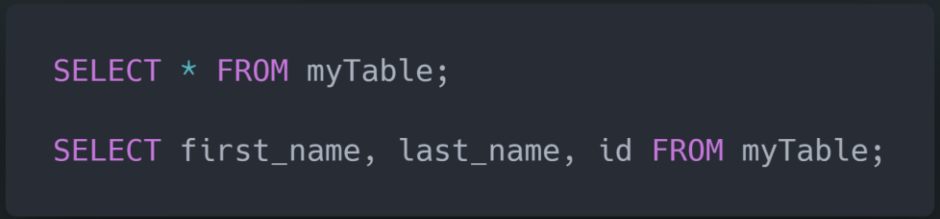
- Create
- Read
- Update
- Delete
- List



A Student Can

retrieve listed data from the database

List



```
SELECT * FROM myTable;

SELECT first_name, last_name, id FROM myTable;
```

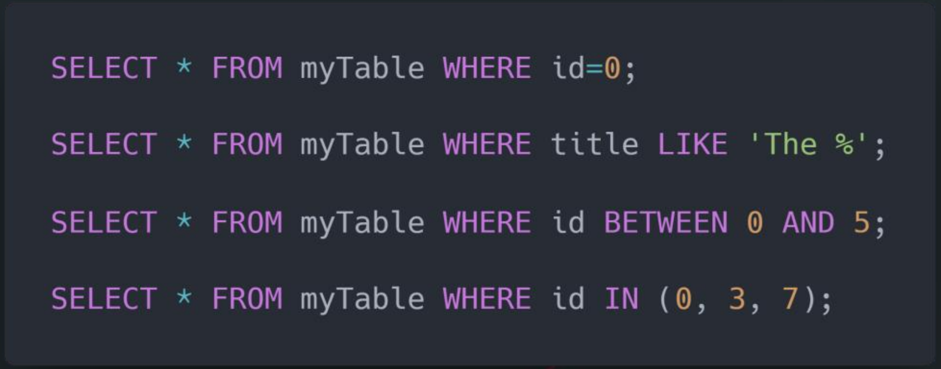
- SELECT
  - What to return
- FROM
  - Where to get it from

A Student Can

retrieve filtered data from the database

Read

Operator	Description
=	Equal
<>	Not equal. **Note:** In some versions of SQL this operator may be written as !=
>	Greater than
<	Less than
>=	Greater than or equal
<=	Less than or equal
BETWEEN	Between an inclusive range
LIKE	Search for a pattern
IN	To specify multiple possible values for a column

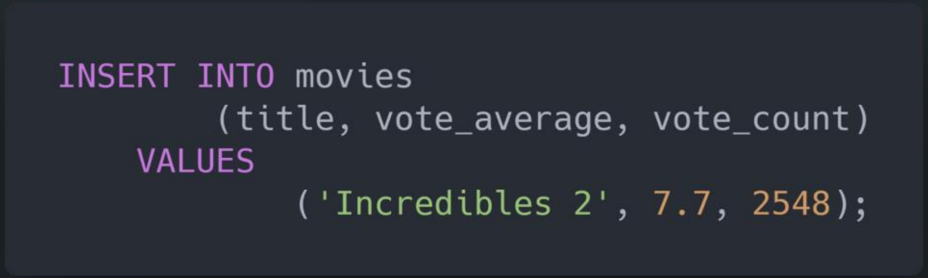


```
SELECT * FROM myTable WHERE id=0;

SELECT * FROM myTable WHERE title LIKE 'The %';

SELECT * FROM myTable WHERE id BETWEEN 0 AND 5;

SELECT * FROM myTable WHERE id IN (0, 3, 7);
```

A Student Can

add data to the database

Create

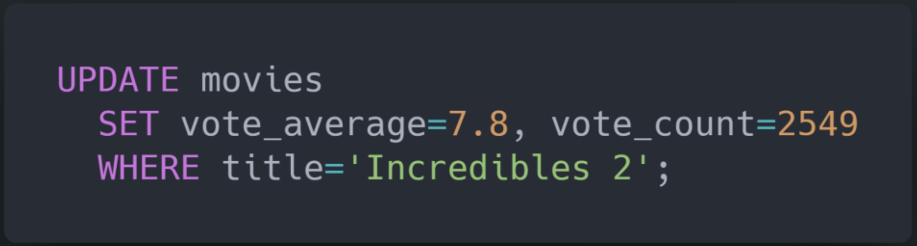


```
INSERT INTO movies
        (title, vote_average, vote_count)
    VALUES
            ('Incredibles 2', 7.7, 2548);
```

- INSERT INTO
  - Table
  - Value names
  - Values

Update

- UPDATE
  - Target table
- SET
  - Values to set
- WHERE
  - Which entry to update

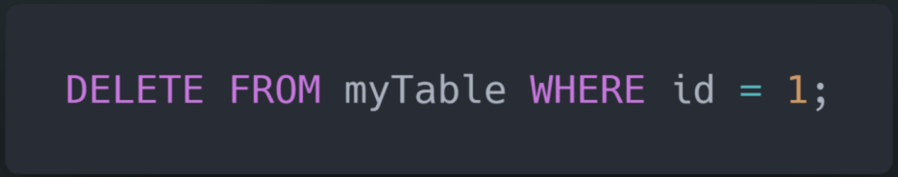


```sql
UPDATE movies
  SET vote_average=7.8, vote_count=2549
  WHERE title='Incredibles 2';
```

A Student Can

remove data from the database

Delete



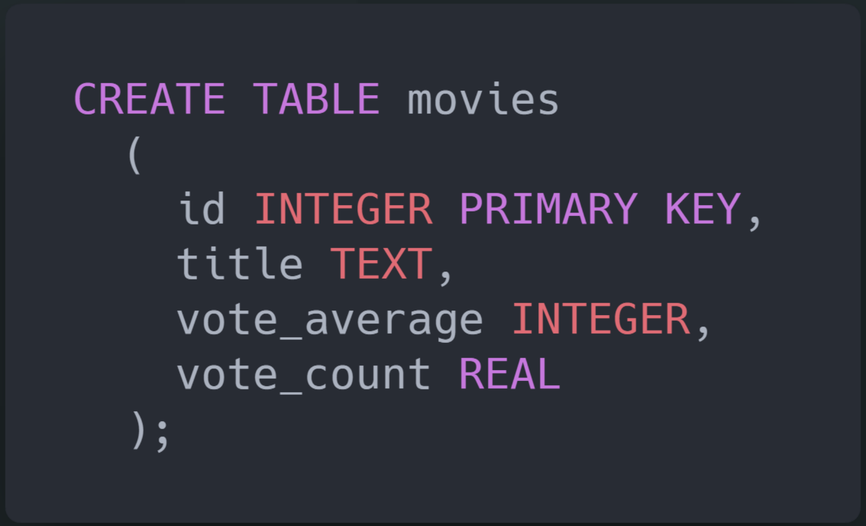
```
DELETE FROM myTable WHERE id = 1;
```

- DELETE FROM
- WHERE

A Student Can

define an SQL Schema

Database Schemas



```
CREATE TABLE movies
  (
    id INTEGER PRIMARY KEY,
    title TEXT,
    vote_average INTEGER,
    vote_count REAL
  );
```

- Structures data
- Limits what types of data can be stored
- Facilitates the power of SQL

Data Types

Type	Meaning
INTEGER	Any number which is no floating point number
REAL	Floating-point numbers (8-Byte IEEE 754 - i.e. double precision)
TEXT	Any String and also single characters (UTF-8, UTF-16BE or UTF-16LE)
BLOB	A binary blob of data

Field Constraints

- Add attributes to fields
- Key to SQL power

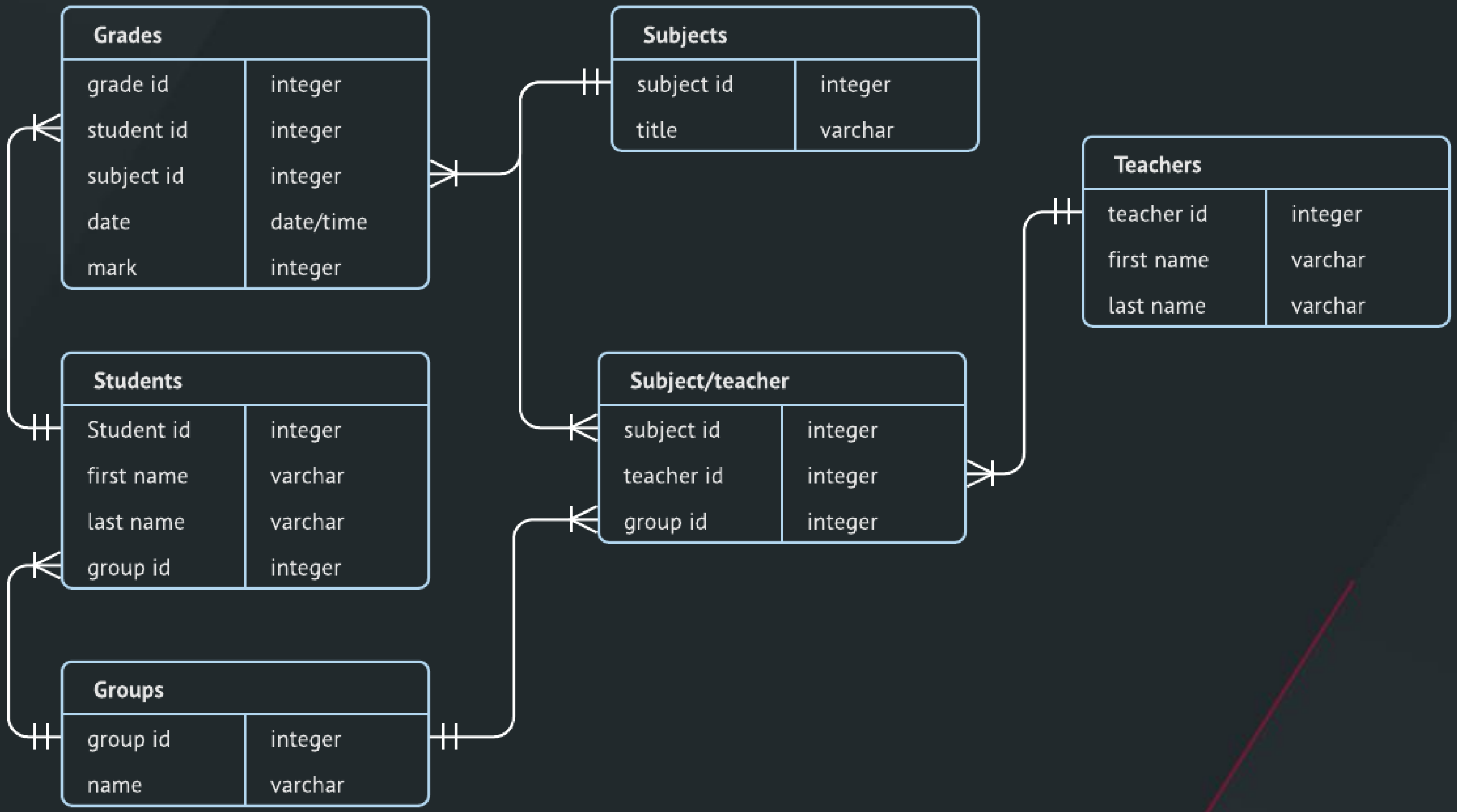
Type	Meaning
NOT NULL	Ensures that a column cannot have a NULL value
UNIQUE	Ensures that all values in a column are different
PRIMARY KEY	A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table
FOREIGN KEY	Uniquely identifies a row/record in another table

Continue if Time

A Student Can

understand and explain the role of a foreign key

Relationships



Grades	
grade id	integer
student id	integer
subject id	integer
date	date/time
mark	integer

Subjects	
subject id	integer
title	varchar

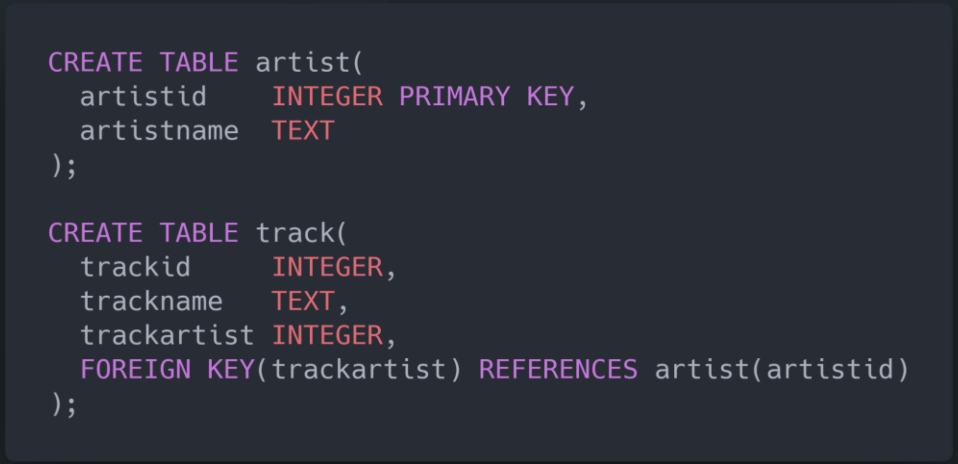
Teachers	
teacher id	integer
first name	varchar
last name	varchar

Students	
Student id	integer
first name	varchar
last name	varchar
group id	integer

Subject/teacher	
subject id	integer
teacher id	integer
group id	integer

Groups	
group id	integer
name	varchar

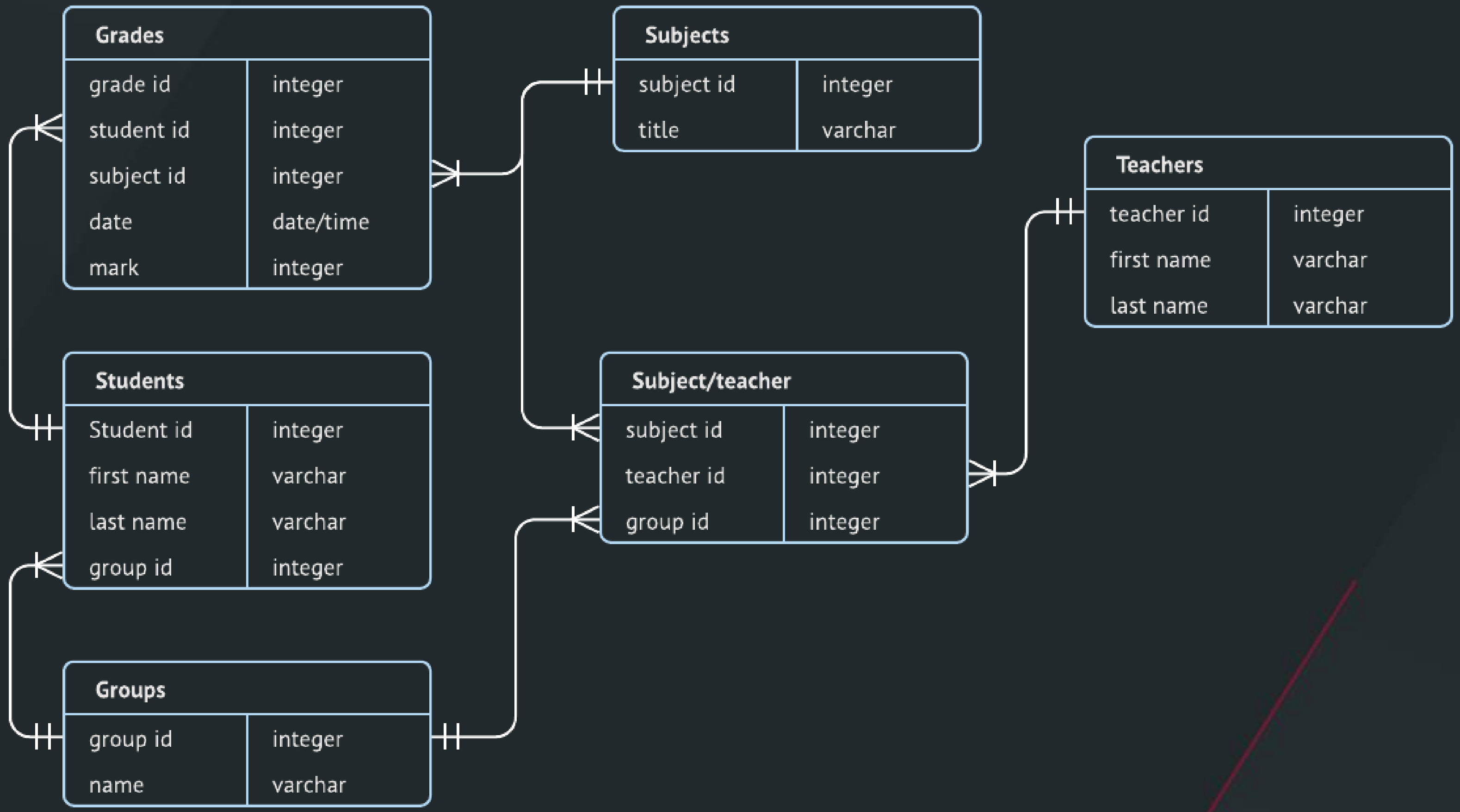
Foreign Keys



```
CREATE TABLE artist(
  artistid    INTEGER PRIMARY KEY,
  artistname  TEXT
);

CREATE TABLE track(
  trackid     INTEGER,
  trackname   TEXT,
  trackartist INTEGER,
  FOREIGN KEY(trackartist) REFERENCES artist(artistid)
);
```

- Key from another entry
- Links Entries together
- Allows for connections between tables and entries

Relationships



Grades	
grade id	integer
student id	integer
subject id	integer
date	date/time
mark	integer

Subjects	
subject id	integer
title	varchar

Teachers	
teacher id	integer
first name	varchar
last name	varchar

Students	
Student id	integer
first name	varchar
last name	varchar
group id	integer

Subject/teacher	
subject id	integer
teacher id	integer
group id	integer

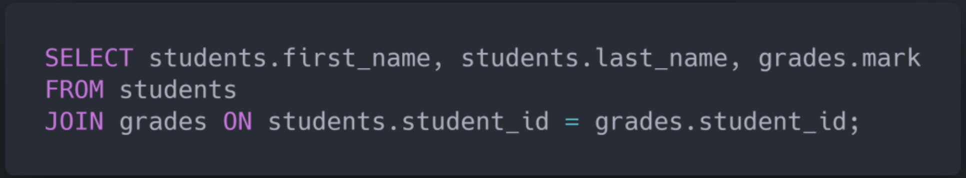
Groups	
group id	integer
name	varchar

A Student Can

combine data from multiple tables into a single table

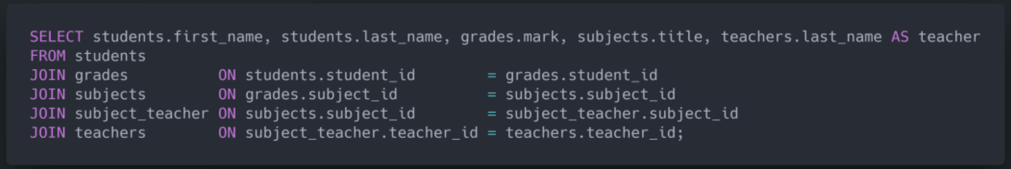
JOINing Tables

- JOIN {table} ON {equivalent values}
- Combine data from multiple tables



```sql
SELECT students.first_name, students.last_name, grades.mark
FROM students
JOIN grades ON students.student_id = grades.student_id;
```

JOINing Many Tables



```sql
SELECT students.first_name, students.last_name, grades.mark, subjects.title, teachers.last_name AS teacher
FROM students
JOIN grades          ON students.student_id        = grades.student_id
JOIN subjects        ON grades.subject_id          = subjects.subject_id
JOIN subject_teacher ON subjects.subject_id        = subject_teacher.subject_id
JOIN teachers        ON subject_teacher.teacher_id = teachers.teacher_id;
```

- SELECT desired fields
- JOIN in order of use